

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (currently amended):

1 A device comprising:

2 a port to receive one or more data streams, each data stream including one or more data
3 frames;

4 a task scheduler coupled to the port, the task scheduler to generate a task identifier for
5 every data frame received;

6 a first queue ~~coupled to the task scheduler~~ to hold task identifiers for which a
7 corresponding data frame is of a first priority type;

8 a second queue ~~coupled to the task scheduler~~ to hold task identifiers for which the
9 corresponding data frame is of a second priority type, the second priority type different
10 than the first priority type;

11 a switch coupled to the first and second queues, the switch configured to retrieve task
12 identifiers from the first queue and the second queue in a fair manner; and

13 a third queue coupled to the switch, the third queue to hold a plurality of task identifiers
14 placed in the third queue by the switch and provide the task identifiers to a processing
15 unit in the order task identifiers were placed in the third queue by the switch;

16 a classifier coupled to the port and to the first queue, the classifier to assign one of a
17 plurality of priority types to every data frame received, the plurality of priority types
18 including the first priority type and the second priority type, wherein the task router is
19 configured the classifier to monitor the first queue for an overflow condition and, if an
20 overflow condition is detected, to reassign data frame priority types from the first priority
21 type to the second priority type to prevent overflow of the first queue. queue; and

22 a task router coupled to the task scheduler, the classifier, the first queue, and the second
23 queue, the task router configured to

24 receive the task identifier from the task scheduler, the task identifier corresponding to
25 a received data frame,

26 receive a priority type from the classifier, the priority type corresponding to the
27 received data frame,

28 place the task identifier in the first queue if the priority type is the first priority type,
29 and

30 place the task identifier in the second queue if the priority type is the second priority
31 type.

2. (cancelled)

3. (previously presented):

1 The device of claim 1 further comprising:

2 a look-up table communicatively coupled to the task scheduler and to the port, the look-
3 up table to provide one of the first priority type and the second priority type to the task
4 scheduler for every data frame received according to the data stream in which the data
5 frame was included.

4. (previously presented):

1 The device of claim 3 wherein one of the first priority type and the second priority type is
2 pre-assigned to the data stream.

5. (previously presented):

1 The device of claim 3 wherein the conversions between priority types and data frame types
2 are dynamically configured in response to usage of the first and second queues.

6.-7. (cancelled)

8. (previously presented):

1 The device of claim 1 wherein the switch is configured to retrieve task identifiers from both
2 the first and second queues in a task retrieval cycle in which at least one task identifier is
3 retrieved from each of the first and second queues such that space in the third queue is
4 allotted equally according to processing time restrictions.

9. (previously presented):

1 The device of claim 1 wherein the switch is configured to retrieve task identifiers with the
2 first priority type until a cumulative processing time requirement for the retrieved task
3 identifiers with the first priority type is substantially equal to a processing time requirement
4 for the task identifiers with the second priority type, and then to retrieve a task identifier with
5 the second priority type.

10. (original):

1 The device of claim 1 wherein the third queue is a shared execution queue from which one or
2 more processing units retrieve task identifiers to process.

11. (currently amended):

1 A method comprising:

2 receiving one or more data streams, each data stream including one or more data frames
3 of one or more data frame types;

4 determining a task priority level for each data frame received;

5 routing each data frame to one of one or more storage queues based on the task priority
6 level of each data frame;

7 retrieving the data frames from the one or more storage queues during a task retrieval
8 cycle according to a fair and weighted processing scheme based on task priority level;
9 and

10 reassigning ~~one or more data frame types to different priority levels to prevent further~~
11 ~~data frames from being stored in a first storage queue if an overflow condition is detected~~
12 ~~in the task priority level for each data frame received prior to routing if an overflow~~
13 ~~condition is detected in a first storage queue and if the task priority level would cause a~~
14 ~~data frame to be stored in the first storage queue, the task priority level being reassigned~~
15 ~~to a task priority level that will cause the data frame to be stored in other than the first~~
16 storage queue.

12. (previously presented):

1 The method of claim 11 wherein the task priority level is determined from one of frame size,
2 echo canceller tail length, codec type, and frame processing requirements.

13. (original):

1 The method of claim 11 wherein the task priority level corresponding to a particular data
2 frame type is pre-configured.

14. (original):

1 The method of claim 11 wherein each storage queue stores data frames of a different task
2 priority level than the other storage queues.

15. (previously presented):

1 The method of claim 11 wherein each data frame type corresponds to a particular processing
2 time requirement for data frames of the data frame type.

16. (original):

1 The method of claim 15 wherein, according to the weighted processing scheme, data frames
2 of approximately equal total processing time restrictions are retrieved from each storage
3 queue in a task retrieval cycle.

17. (cancelled)

18. (previously presented):

1 The method of claim 11 further comprising:
2 placing a plurality of the retrieved data frames into an execution queue to be processed.

19. (currently amended):

1 A method comprising:
2 receiving one or more data streams, each data stream including one or more data frames
3 of one or more data frame types;
4 determining a task priority level for each data frame received;
5 assigning a unique task identifier to each received data frame;
6 storing each task identifier to one of multiple storage queues according to the task priority
7 level of the corresponding data frame;
8 retrieving task identifiers from the one or more storage queues during a task retrieval
9 cycle according to a weighted processing scheme based on task priority levels; and
10 reassigning ~~one or more data frame types to different priority levels~~ the task priority level
11 ~~for each data frame received prior to storing each task identifier if an overflow condition~~
12 ~~is detected in a first storage queue and if the task priority level would cause a task~~
13 ~~identifier to be stored in a first storage queue, the task priority level being reassigned to a~~
14 ~~task priority level that will cause the task identifier to be stored in other than~~ to prevent
15 ~~further data frames from being stored in a first storage queue if an overflow condition is~~
16 ~~detected in the first storage queue.~~

20. (previously presented):

1 The method of claim 19 wherein the task priority level is determined from one of frame size,
2 echo canceller tail length, codec type, and frame processing requirements.

21. (previously presented):

1 The method of claim 19 wherein each data frame type corresponds to a particular processing
2 time requirement for data frames of the data frame type.

22. (original):

1 The method of claim 21 wherein, according to the weighted processing scheme, task
2 identifiers corresponding to data frames of approximately equal total processing time
3 requirements are retrieved from each storage queue in a task retrieval cycle.
4

23. (previously presented):

1 The method of claim 19 further comprising:
2 placing a plurality of the retrieved task identifiers into an execution queue to be processed.

24. (currently amended):

1 A machine-readable medium having one or more instructions for scheduling processing tasks,
2 which when executed by a processor, causes the processor to perform operations comprising:
3 receiving one or more data streams, each data stream including one or more data frames
4 of one or more data frame types;
5 determining the task priority level for each ~~of the data frames~~ frame received;
6 routing each data frame to one of one or more storage queues based on the task priority
7 level of each data frame; and
8 retrieving the data frames from the one or more storage queues during a task retrieval
9 cycle according to a fair and weighted processing scheme based on task priority ~~levels~~
10 level; and
11 ~~reassigning one or more data frame types to different priority levels to prevent further
12 data frames from being stored in a first storage queue if an overflow condition is detected
13 in the task priority level for each data frame received prior to routing if an overflow
14 condition is detected in a first storage queue and if the task priority level would cause a
15 data frame to be stored in the first storage queue, the task priority level being reassigned~~

16 to a task priority level that will cause the data frame to be stored in other than the first
17 storage queue.

25. (currently amended):

1 The machine-readable medium of claim 24 wherein the task priority level is determined
2 from one from one of frame size, echo canceller tail length, codec type, and frame processing
3 requirements.

26. (original):

1 The machine-readable medium of claim 24 wherein each storage queue stores data frames of
2 a different task priority level than the other storage queues.

27. (previously presented):

1 The machine-readable medium of claim 24 wherein each data frame type corresponds to a
2 particular processing time requirement for data frames of the data frame type.

28. (original):

1 The machine-readable medium of claim 27 wherein, according to the weighted processing
2 scheme, data frames of approximately equal total processing time are retrieved from each
3 storage queue in a task retrieval cycle.

29. (cancelled)

30. (previously presented):

1 The machine-readable medium of claim 24 further comprising:
2 placing a plurality of the retrieved data frames into an execution queue to be processed.